SETSCI conference proceedings
SETSCI Conference Proceedings
14 , 84-88, 2022

6th International Symposium on Innovative Approaches in Smart Technologies
December 8-10, 2022, Turkey

ISAS WINTER-2022

https://doi.org/10.36287/setsci.5.2.018

Uçak Görüntülerinin Sınıflandırılmasında Farklı Yapay Zekâ Algoritmalarının Performansı

Özgün Devrim KILIÇ[1], Mustafa Emre AYDEMİR[2]*, Pınar ÖZTÜRK ÖZDEMİR[3]
[1]Milli Savunma Üniversitesi, İstanbul, Türkiye, dkilic1@hho.msu.edu.tr, ORCID: 0000-0003-2146-996X
2*İstanbul Esenyurt Üniversitesi, İstanbul, Türkiye, mustafaaydemir@esenyurt.edu.tr, ORCID: 0000-0002-9285-5115
3Milli Savunma Üniversitesi, İstanbul, Türkiye, oozdemir33@hho.msu.edu.tr, ORCID: 0000-0002-9032-8987

Özet – Günümüzde yapay zekâ uygulamaları ile nesne tespiti birçok alanda kullanılmaktadır. Yapay zekâ ile görüntü işleme teknolojisinin ve veri setlerinin gelişmesiyle birlikte birçok matematiksel model ve algoritma nesne tespitinde kullanılmaktadır. Yapay zekâ algoritmasının başarısı ve verimliliği de kullanıcı için aynı oranda önem arz etmektedir. Bu çalışmada farklı zeminlerde ve açılardan çekilmiş uçak fotoğrafının OpenCv kütüphanesinde farklı algoritmalarla tespit edilmesi ve algoritmaların tespit etmedeki başarılarının kıyaslanması amaçlanmıştır. Bu tespit aşamasında Haarcascade ve YOLO algoritmaları kullanılmıştır. Çekilen fotoğraflar etiketlenmiş ve veri seti oluşturulmuştur. Bir sonraki aşamada Haarcascade modeli ile ortam oluşturulmuş ve fotoğraflar eğitilmiştir. İkinci aşamada YOLO algoritması üzerinden ortam oluşturulup fotoğraflar eğitilmiştir. Bu iki algoritma ile ayrı ayrı nesne tespiti yapılmış olup sonuçlarının kıyaslanması için Confisuon Matrix oluşturulmuştur. Kıyaslama sonrasında metrikler, grafikler ve tablo halinde değerlendirilmiştir.

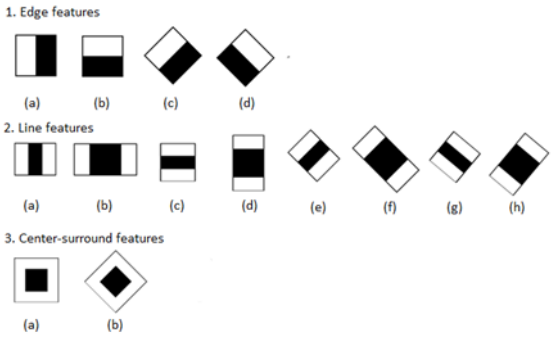
Keywords: Yapay zekâ, algoritma kıyaslaması, HaarCascade, YOLOv5

I. GİRİŞ

Görüntü işleme teknolojisi ilk başta görüntülerin kalitesini arttırmak amaçlı kullanılmaktaydı. Gelişen teknolojiyle birlikte görüntü işleme uygulamalarında yapay zekânın kullanılması zorunlu hale gelmiştir. Yapay zekâ kullanımının makine öğrenmesi tekniği başta olmak üzere, bu tekniğin derin öğrenme, yapay sinir ağları gibi alt kümelerde görüntü işleme teknolojisinde yararlanılması ilerleyen süreçlerde beraberinde gelmiştir. Bu sayede görüntü işleme süreci kısalmış ve verimlilik arttırılmıştır.

Yapılan literatür araştırmalarında yüz tespiti konusu ağırlıklı olmak üzere birçok yapay zekâ uygulaması kullanılmıştır. Bu bildiride ise uçak görüntüleri kullanarak hangi yapay zekâ algoritmasının bu nesneleri daha doğru tespit edilebileceği konusu üzerinde çalışılmıştır.

İlk olarak Paul Viola ve Michael Jones tarafından önerilen Haar ardışık sınıflandırıcı görüntünün boyutundan ve konumundan bağımsız olarak nesne tespit edebilen makine öğrenmesi tabanlı algoritmalardan biridir [**1**]. Bu algoritmanın birincil faydası hesaplama hızının yüksek olmasıdır.[**2**] Haarcascade sabit boyutlu bir çerçeveyi pozitif görüntü üzerinde birçok ölçekte adım adım tarayarak, çerçeve içerisindeki siyah bölgedeki piksel değerleri toplamıyla, beyaz bölgedeki piksel değerleri toplamlarından bu bölgede nesne vardır veya nesne yoktur diyerek tespit etme mantığına dayanır.[**3**].



Resim 1. Ortak Haar Özellikleri [4]

Bu bildiride sunulan çalışmada Open Cv 4.5.4 kütüphanesini tanıttıktan sonra bu algoritma kullanılarak çektiğimiz uçak fotoğraflarının pozitif ve negatif örneklerini oluşturulmuştur. Ardından bu örneklerin dosya yollarının bulunduğu text dosyası hazırlanmış pozitif olanları işaretlenmiştir. Sonraki aşamada sınıflandırıcı eğitilmiş ve sınıflandırıcı kullanılarak nesne algılanmıştır.[**3**]

İkinci aşamada Uçak resimlerini YOLO algoritması kullanılarak tespit etme amaçlanmıştır. YOLO konvolüsyonel sinir ağlarını kullanarak(CNN) nesne tespiti yapmayı amaçlayan bir algoritmadır. Açılımı ”You Only Look Once” yani “Sadece bir kere bak” anlamına gelir. Deforme Parça Modelleri (DPM) Evrişimli Sinir Ağları (RCNN) modellerine göre YOLO görüntüyü bir seferde sinir ağından geçirdiği için nesnenin tipini ve görüntüdeki konumunu tespit etmede daha iyi bir performans gösterir. Gerçek zamanlı nesne tespitinde saniyede 45 kare yakalama(FPS) özelliğine sahiptir. [**4**]Fakat yapılan çalışmalarda küçük boyutlu ve çok yakın konumlanmış nesnelerde yanlış konum verebilme ihtimali bulunmaktadır.

YOLO algoritması görüntüyü ağ gözleri(mesh) denilen bölgelere ayırır ve bölge içerisinde ‘sınır kutuları’ çizerek

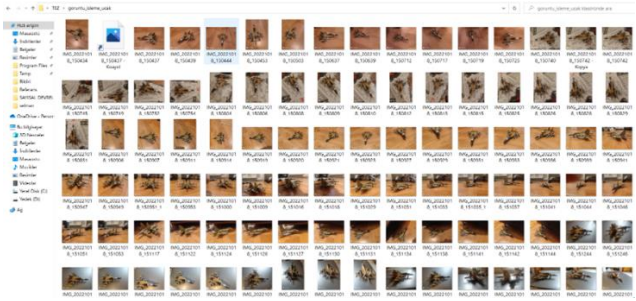
tespit etmek istediğimiz objenin orada olup olmadığı ile ilgili hesap yapar. Bu hesap sonrasında çizdiği sınır kutusunun doğruluğu hakkında bize 0-1 arasında güven skoru verir [**5**].

Çalışmamızda uçak resimlerini tespit etmek için ayrı ayrı Haarcascade ve YOLOv5 algoritmaları kullanılmış olup tahmin edilen ve gerçekleşen değerleri "Confusion Matrix" kullanılarak değerlendirilmiştir. Bunun yanı sıra doğruluk kesinlik ve duyarlılık gibi metrikler de kullanılarak karşılaştırmaya eklenmiştir. Bildirinin 2. kısmında algoritmaları kıyaslarken kullanılan Materyal ve Metod hakkında bilgi verilmiştir. Bildirinin 3. kısmında Sonuçlar yer almaktadır.

II. METODOLOJİ VE BULGULAR

Yapay zekâ algoritmaları Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz işlemciye, NVIDIA GeForce RTX 3060 grafik kartına ve 16 GB RAM'e sahip ve 64-bit Windows 10 işletim sistemi kurulu bir bilgisayarda gerçekleştirilmiştir. Derin öğrenme eğitimi Phyton 3.8 versiyonu Numpy kütüphanesi ve Phycharm geliştirme ortamı kullanılarak uygulanmıştır. Fotoğraf çekimi için 12MP görüntü kalitesine sahip Xiaomi Mi8 telefonla yaklaşık 400 adet, farklı açılardan ve ortamlardan alınmış uçak fotoğrafı kullanılarak veri seti oluşturulmuştur. Daha sonrasında veri seti etiketleme işlemi .xml uzantılı dosyalar kullanılarak oluşturulmuş ve eğitime geçilmiştir.

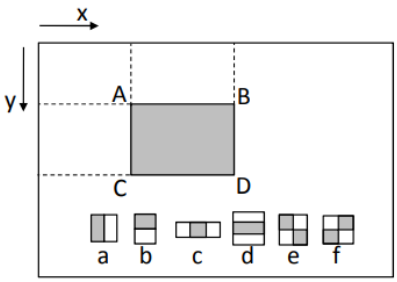
Eğitim iki farklı algoritma üzerinden gerçekleştirilmiş olup sonuçları kıyaslanmış hangi algoritmanın daha başarılı olduğu değerlendirilmiştir.



Resim 2. Uçak Görüntüleri Veri Seti

A. *HaarCascade Algoritması*

Bu algoritma belirli ölçeklerde oluşturulmuş zayıf sınıflandırıcılar(Feature) pozitif resimler üzerinde gezdirilerek aydınlık bölgeleri karanlık bölgelerden ayırt etme mantığına dayanır. Zayıf sınırlandırıcılar resimleri adım adım tarayarak negatif resimler üzerinde tespit yapamamakta, pozitif resimler üzerinde ise nesne tespit ederek bunların üzerinde piksel değerlerini integralle toplamaktadır. [**3**] Bu süreç görüntü işleme sırasında bilgisayarın tekrar tekrar toplama işlemi yapmasına engel olmaktadır. Ayrıca resimlerin taranması esnasında her çerçeve baştan tüm görüntüleri taramak yerine önceki görüntülerde eşleşmiş kısımları atlayarak zaman kazanmaktadır. Bu şekilde [**3**] Haar Cascade algoritmasının çalışma hızı buna bağlı olarak işlemci gücü, örnekleme sayısı ve eğitim biçimine göre farklılık göstermektedir. HaarCascade'ın doğruluk oranının yanı sıra nesneyi tespit etme hızı bu faktörlerin de iyileştirilmesiyle arttırılmaktadır



Resim 3. İntegral Görüntünün Çalışma Şekli [6]

Haar Cascade sınıflandırıcısının eğitimi için pozitif ve negatif görüntüler oluşturulmuştur. Pozitif görüntüler içerisinde uçak nesnesinin bulunduğu; negatif görüntüler uçak nesnelerinin bulunmadığı görüntüleri içerir. Pozitif ve negatif görüntülerin sayısının fazla olması sınıflandırıcının daha doğru nesne tespit etmesine yardımcı olmaktadır. [**3**] Pozitif ve negatif resimlerin .txt uzantılı dosyalarını oluşturulduktan sonra pozitif resimleri haarcascade sınıflandırıcısında kullanabilmek için gri skalaya çevirmemiz, uçağın bulunduğu kısmı kırpmamız ve bir vektör dosyası oluşturmamız gerekmektedir. [**3**] Haar Cascade algoritmamızı eğittikten sonra oluşan .xml uzantılı dosyayı opencv'deki kütüphanesinde kullanılan projeyle birlikte çalıştırarak Haar Cascade'ın görüntü üzerindeki uçağı algılaması sağlanmaktadır.

B. *YOLO Algoritması*

YOLO algoritması diğer algoritmalardan farklı olarak gerçek zamanlı nesne tespit etme işlemini daha hızlı ve tek seferde sinir ağından geçirerek yapabilmektedir. Ayrıca mAP (kesinlik) değerinin de yüksek olması YOLO'yu diğer algoritmalardan daha avantajlı hale getirmiştir. COCO veri setinde yapılan karşılaştırmada YOLO'nun rakiplerine karşı mAP-50 üzerinde rakiplerine göre zaman ve doğruluk parametrelerinde göre değerlendirmede çok başarılı olduğunu gözlemleyebiliyoruz.[**7**] YOLO algoritmasının diğer algoritmalara göre avantajı FPS(frame per second,saniyede yakaladığı kare) miktarının yüksek olmasıdır.

Bu algoritma temelde tüm resmi AxA'lık ağ gözlerine ayırarak, her bir ağ gözü içerisinde tespit edilecek sistemin orta noktasının koordinatlarını, nesnenin genişliğini, yüksekliğini ve nesnenin türünü tespit etmeye çalışır. Tahmin ettiği noktalar üzerinden bir "Tahmin Vektörü" oluşturur. Tahmin vektörü içerisinde:

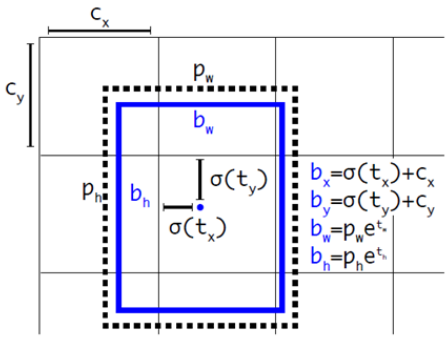
Güven Skoru: Bulmak istediğimiz nesnenin göz ağı içerinde bulunma ihtimalidir. Kutu güven skoru ile bağlı sınıf olasılığının çarpımından oluşur.(0-1 arasındadır.)

$$Güven\ Skoru = Kutu\ Güven\ Skoru * Bağlı\ Sınıf\ Olasılığı \quad (1)$$

$$Kutu\ Güven\ Skoru = P * IoU \quad (2)$$

P: Kutunun nesneyi kapsayıp kapsamadığının ihtimali.
IoU: Yer doğrusu ile kutu arasındaki IoU değeri.
B_X: Nesne orta noktası X koordinatı
B_Y: Nesne orta noktası Y koordinatı.
B_W: Nesne genişliği
B_H: Nesne yüksekliği
Bağlı Sınıf Olasılığı: Modelde kaç sınıf varsa o kadar sayıda tahmin değeridir.(0 ila 1 arasında) [**7**]

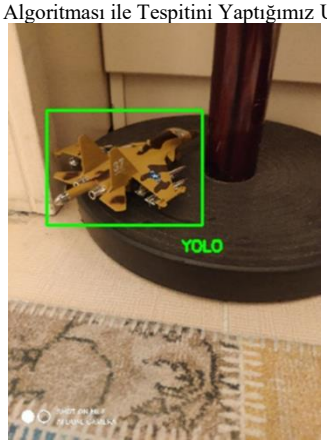
Hiçbir nesne olmayan ağ gözlerinde bağlı sınıf olasılığı sıfır olmalıdır. Doğal olarak güven skoru da sıfır olacağı için nesne tespit etmek söz konusu olmayacaktır. Yaptığımız tanımlamaya göre görüntüdeki her bir ağ gözü yalnızca bir nesne tespit edebilmektedir. Bu sayıyı arttırabilmek için "Anchor Box" metodunu uygulamak gerekmektedir.[**9**] Ağ gözü içerinde farklı farklı geometrik kalıplar kullanılarak bir göz içerisinde daha fazla nesne tespit edebilmemizi sağlamaktadır. Bu da tahmin vektörünü her bir "Anchor Box" için ayrı ayrı güven skoru, orta nokta, koordinat ve sınıf olasılığı hesaplamasına sebep olmaktadır. Bu yöntem sayesinde anchor boxların sayısı arttırılarak bir ağ gözü içerinde birden fazla nesne olması durumunda algoritmanın nesneleri tespit etmede yanılmasının önüne geçilmektedir.[**11**]



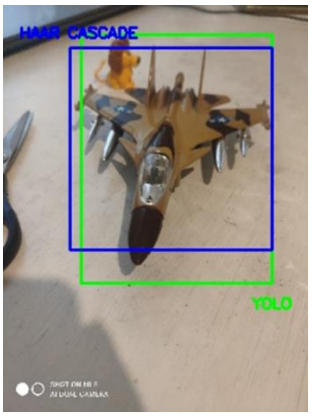
Resim 4. Anchor Box ile Boyut ve Konum Tahmini Yapan Sınır Kutusu [8]

Uçak görüntülerinin tespit edilmesinde bazı durumlarda bir göz içerisinde YOLO birden fazla kere uçağı tespit etmiştir. Bu da gereksiz birçok kutu oluşumuna sebep olmuştur. Bunun önüne geçmek için algoritma güven skoru belli bir değerin altında olan kutuları atarak en yüksek başarılı tahmin yapan kutuyu tutmuştur.

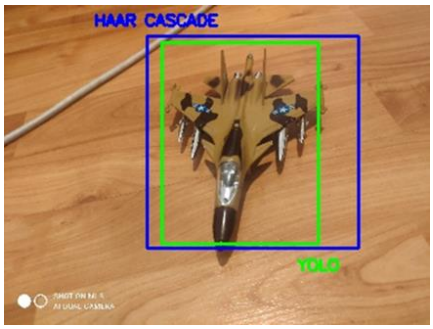
Resim 5. YOLO Algoritması ile Tespitini Yaptığımız Uçak Görüntüsü-1



Resim 6. YOLO Algoritması ile Tespitini Yaptığımız Uçak Görüntüsü-2



Resim 7. YOLO Algoritması ve HaarCascade ile Birlikte Tespitini Yaptığımız Uçak Görüntüsü-1



Resim 8. YOLO Algoritması ve HaarCascade ile Birlikte Tespitini Yaptığımız Uçak Görüntüsü-2

C. Confisuon Matrix ve Bazı Metrikler:

Algoritma kıyaslama aşamasında sadece Doğruluk metriği üzerinden kıyaslama yapmamak gerekmektedir. Bu noktada Confisuon Matrisi nesnelerin olması gereken sınıflarla, tahmin ettiğimiz sınıfları kıyaslayan ve tahminlerde yapılan hataların nerelerde olduğunu bulan matristir. Matrisimizde True Positive ve True Negative modelin doğru olarak tahmin edildiği, False Positive ve False Negative ise modelin yanlış olarak tahmin edildiği kısımlardır.[**10**]

True Positive(TP): Uçak nesnesinin varlığı durumunu doğru bilmek.

True Negative(TN): Uçak nesnesinin olmamasını durumunu doğru bilmek.

False Pozitive(FP): Uçak nesnesinin olmaması durumunu yanlış bilmek.

False Negative(FN): Uçak nesnesinin varlığı durumunu yanlış bilmek.

Yolov5 10 epoch	Airplane	Background
Airplane	99	0
Background	1	100

ACTUAL			
Positive	*Negative*		
True Positive	False Positive	*Positive*	PREDICTION
False Negative	True Negative	*Negative*	

Resim 7. Confusion Matrix

Accuracy(Doğruluk): Doğru tahmin edilen uçak görüntülerinin toplam uçak görüntülerine oranlanarak bulunur. Tek başına değerlendirma yapmak için yeterli değildir.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \quad (3)$$

Recall(Duyarlılık): Uçak nesnesini doğru tespit etme oranıdır. Bu metrikte uçağın varlığını tespit etmek,uçağın olmadığı duruma uçak var demekten daha kritiktir.

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Precision(Kesinlik): Uçak var tespiti yaptığımız resimlerden kaçında uçak olduğunu tahmin etme oranıdır. Bu metrikte uçağın varlığını kesin olarak söyleme ihtimali artmaktadır.

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

F1 Skoru: Recall ve Precision değerlerinin harmonik ortalamasını göstermektedir. Bunu yaparak uç noktalardaki değerleri devre dışı bırakarak sağlıklı bir ortalama almaktır.

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (6)$$

III. TABLO VE GRAFİK SONUÇLARI

Bu bölümde YOLOv5 ve HaarCascade ile yaptığımız nesne tespit çalışmalarının karşılaştırmalı sonuçları ve grafikleri yer almaktadır. Algoritmamız YOLOv5 için 7-10-15 epoch(eğitim turu) sayısı ile ayrı ayrı sonuçlar değerlendirilmiştir. Derin öğrenmede nesneyi tanıma işlemini çözecek en uygun ağırlık değerleri epoch(eğitim turu) ile hesaplanır. Bu da epoch sayısı arttıkça başarı da artacaktır. Fakat epoch sayısını çok yüksek tutmak da belli zaman sonra öğrenmeyi güçleştireceği için eğitim süresini de uzatacaktır.

Tablo 1. YOLOv5 epoch 7 için Confusion Matrix

.

Yolov5 7 epoch	Airplane	Background
Airplane	89	0
Background	11	100

Class	Images	Instances	P	R	mAP50	mAP50-95: 100%
all	70	70	0.996	1	0.995	0.763
airplane	70	70	0.996	1	0.995	0.763

Resim 8. YOLOv5 epoch 7 için Metrik Sonuçları

Tablo 2. YOLOv5 epoch 10 için Confusion Matrix

Class	Images	Instances	P	R	mAP50	mAP50-95: 100%
all	70	70	0.944	0.971	0.987	0.615
airplane	70	70	0.944	0.971	0.987	0.615

Resim 8. YOLOv5 epoch 7 için Metrik Sonuçları

Tablo 3. YOLOv5 epoch 15 için Confusion Matrix

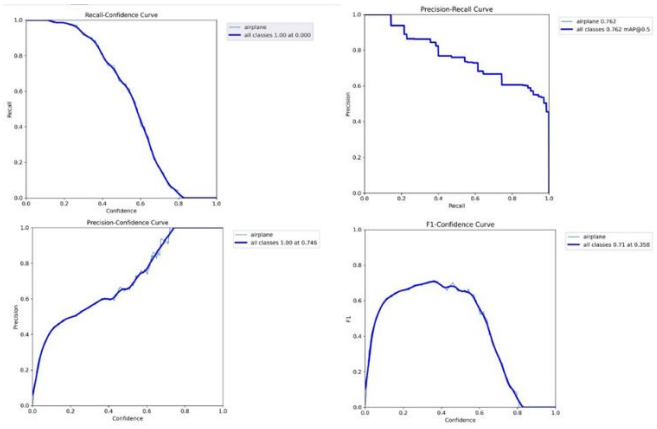
Yolov5 15 epoch	Airplane	Background
Airplane	100	0
Background	0	100

Class	Images	Instances	P	R	mAP50	mAP50-95: 100%
all	70	70	0.996	1	0.995	0.763
airplane	70	70	0.996	1	0.995	0.763

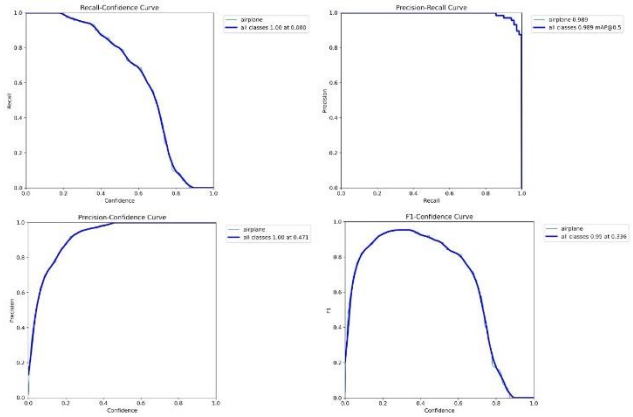
Resim 9. YOLOv5 epoch 15 için Metrik Sonuçları

Tablo 4. Haarcascade için Confusion Matrix

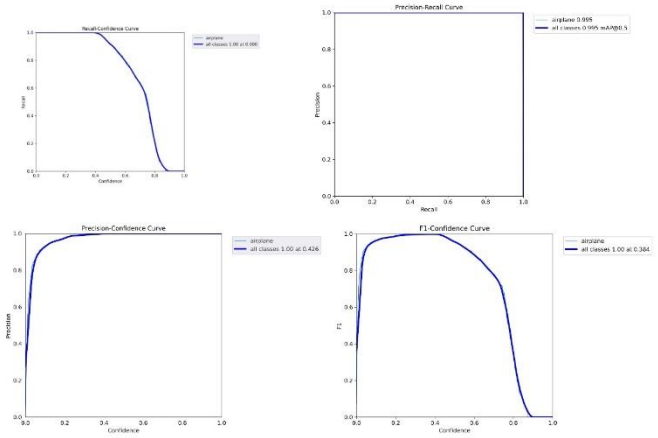
Haar Cascade	Airplane	Background
Airplane	72	0
Background	28	100



Resim 10. YOLOv5 epoch 7 için Metriklerin Grafiklerle Gösterimi



Resim 11. YOLOv5 epoch 10 için Metriklerin Grafiklerle Gösterimi



Resim 12. YOLOv5 epoch 15 için Metriklerin Grafiklerle Gösterimi

III. ÇÖZÜM

Bu çalışma sonucunda yaklaşık 400 fotoğraf üzerinden iki farklı yapay zekâ algoritmasıyla uçak tespiti yapılmaya çalışılmış olup YOLO'nun HaarCascade algoritmasına göre daha doğru sonuç verdiği ve kullanışlı olduğu tespit edilmiştir. YOLO algoritmasını kendi içerinde Epoch sayısına göre kıyasladığımızda ise Epoch saysını 15'e kadar çıkardığımızda doğruluk, duyarlılık, kesinlik ve F1 metriklerinin ve grafiklerinin en doğru tahminleri yaptığı söylenebilmektedir.

REFERANSLAR

[1] Viola P., Jones M., "Rapid object detection using a boosted cascade of simple features", Accepted Conference on Computer Vision and Pattern Recognition, 2001.

[2] Pavani K.,Siriramya P., "Comparison of KNN, ANN, CNN and YOLO algorithms for detecting the accurate traffic flow and build an Intelligent Transportation System", 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Uttar Pradesh, 2022

[3] Koç T.,"HaarCascade Eğitimi", 3 Kasım 2022, https://talhakoc.net/haar-cascade-egitimi/

[4] Rastogi A.,Ryuh B.S.," Teat detection algorithm: YOLO vs. Haar-cascade", School of Mechanical Systems Engineering, Chonbuk National University, January 10, 2019

[5] Görsel Analiz,"YOLO nedir?", 3 Kasım 2022, http://gorselanaliz.com/yolo-nedir/

[6] Zhang C.,Zhang Z., "A Survey of Recent Advances in Face Detection",Microsoft Research Microsoft Corporation, 2010

[7] Mesci Y.,"YOLO Algoritmasını Anlamak" 4 Kasım 2022, https://medium.com/deep-learning-turkiye/yolo-algoritmasını anlamak-290f2152808f

[8] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger.In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017.

[9]Vinh T.,Anh N. "Real-Time Face Mask Detector Using YOLOv3 Algorithm and Haar Cascade Classifier", International Conference on Advanced Computing and Applications (ACOMP), 2020.

[10]Öğündür G.,"Doğruluk(Accuracy), Kesinlik(Precision), Duyarlılık(Recall) ya da F1Score",5 Kasım 2022, https://medium.com/@gulcanogundur/do%C4%9Fruluk-accuracy-kesinlik-precision-duyarl%C4%B1l%C4%B1k-recall-ya-da-f1-score-300c925feb38

[11] Redmon J.,Farhadi A. "YOLOv3: An Incremental Improvement" University Of Washington